Published in the Proceedings of 19th Annual Software Engineering Workshop, NASA Goddard Space Flight Center, Greenbelt, MD, November 30-December 1, 1994.

A Process Improvement Model for Software Verification and Validation

John Callahan[1] *George Sabolish*
NASA Software IV&V Facility
West Virginia University

Abstract

We describe ongoing work at the NASA Independent Verification and Validation (IV&V) Facility to establish a process improvement model for software verification and validation (V&V) organizations. This model, similar to those used by some software development organizations, uses measurement-based techniques to identify problem areas and introduce incremental improvements. We seek to replicate this model for organizations involved in V&V on large-scale software development projects such as EOS and Space Station. At the IV&V Facility, a university research group and V&V contractors are working together to collect metrics across projects in order to determine the effectiveness of V&V and improve its application. Since V&V processes are intimately tied to development processes, this paper also examines the repercussions for development organizations in large-scale efforts.

1 Introduction

In effort to improve the quality of software products in safety-critical and high-risk projects, many organizations employ verification and validation (V&V) techniques to detect and correct errors made during the development process. Verification involves analyzing software products after each major development stage to ensure that the product agrees with the specification established prior to that stage. Validation involves ensuring that the products after each stage agree with the original specifications. Although validation is traditionally performed only at later stages (i.e., testing) with respect to requirements, we employ the broader definition.

A specific application of V&V can be characterized along three dimensions: orientation, scope, and independence. First, V&V activities can focus on either the software development process or the products produced by that process. Most V&V activities, however, perform a combination of both process-oriented and product-oriented analysis. Second, the scope of V&V activities can range from being comprehensive across all development phases, to being limited to specific subsystems and process stages. Finally, V&V activities can be embedded within or independent of a development effort. Independence can vary over levels of technical, managerial, and financial control [10].

Regardless of its organization, however, all V&V organizations are charged with detecting (and sometimes correcting) errors in software products and processes as early as possible in the development life-cycle. This implies that effective techniques must be employed that help find the most critical

---

[1]This work is supported by NASA Cooperative Agreement NCCW-0040 under the supervision of the NASA Headquarters Office of Safety and Mission Assurance (Code Q) at the Independent Software Verification and Validation (IV&V) Facility in Fairmont, West Virginia.

problems in early phases. Clear correlation must be established between these early errors and their consequences later in the development life-cycle. Otherwise, such problems can be dismissed as false warnings or non-critical.

This paper describes ongoing work at the NASA IV&V Facility to develop a process improvement model for software V&V organizations. Our effort involves establishing a framework for iterative measurement and ongoing improvement of a V&V organization's ability to find critical errors early and more accurately estimate costs and benefits of V&V. Although our model is still evolving, we are working with V&V contractors to assess the effectiveness the approach on existing projects.

2 Related Work

There is a limited amount of empirical evidence on V&V in practice, but most of the research on V&V has focused on (1) determining the cost effectiveness of V&V relative to the cost of the overall software development effort; and (2) developing methods for identifying errors as early as possible in the software development life-cycle. First, the cost effectiveness of V&V has been found to depend heavily on many factors including project size, expected lifetime of the software, volatility of requirements, and the expertise of development and V&V personnel. Secondly, even if these factors warrant the use of V&V, it is most important to determine how much, when, and what types of V&V to apply in each project. Effective methods for detecting critical errors must exist to enable an adequate appraisal of what the V&V effort saved in a project [1].

One of the most comprehensive studies of V&V [2] concludes that V&V is highly cost effective if applied early in the life-cycle of large, complex software projects. This study, conducted by NASA/JPL, consists of a survey of over 80 papers and related projects that include both quantitative and qualitative assessments of V&V cost effectiveness. The JPL study strongly suggests that many projects found V&V to be cost effective because the cost to correct latent errors grows exponentially in later life-cycle phases. According to several key papers in the JPL study [3,4,5], V&V can find errors early and avoid the costs of fixing latent errors. Overall, the JPL study suggests that V&V can pay for itself if started in the requirements phase, but also that V&V can negatively impact a project if started late.

In addition, several papers examined in the JPL study conclude that V&V also has benefits such as significantly reduced software maintenance costs [3,6,7]. These studies find that V&V more than pays for itself in projects with long lifetimes due not only to increased reliability but also to decreased maintenance costs. They suggest that V&V increases external management and technical visibility that is essential in long-term projects where personnel turnover is high and requirements are volatile.

Other research has focused on developing effective V&V methods for detecting errors. Many of these methods are specific to software application domains, development processes, and specification techniques. Some methods have proven nominally effective and even ineffective when applied incorrectly [8,9]. For example, a formal verification of code is considered too costly in low-risk projects. Although a formal verification would increase reliability, it would not be cost effective relative to the impact of errors. In this case, the cost of finding the errors exceeds the cost of the error occurring plus the cost of fixing the problem. The high costs of formal verification, however, can be justified in some safety-critical applications where the costs of failure can be catastrophic.

Finally, there are several reports that advocate the use of V&V based on case studies and expert opinion [7,10]. For example, the NRC assessment of Space Shuttle flight software development [10] strongly advocates the continued use of V&V on Shuttle and other large NASA projects. The NRC committee advises that independent V&V can be highly cost effective and useful in avoidance of catastrophic incidents in large projects because it provides visibility into highly complex interactions (often informal) between large numbers of contractors. Because of the informal nature of many of these interactions and

the high turnover of personnel in large projects, an independent V&V contractor can provide continuity over the long-term on large projects and provide management and technical visibility to the customer.

3 Process Improvement for V&V

We are engaged in establishing a process improvement model for V&V organizations at the NASA IV&V Facility [11]. Our objective is to establish criteria for measuring V&V activities, measure on-going V&V projects, and suggest incremental improvements to both product analysis and a V&V process. Although our collaborations are primarily with highly independent V&V groups, small V&V groups are also involved within specific projects.

To accomplish our objective, we are building a process improvement model for V&V based on measurement of products and processes from both development and V&V efforts. Our proposed model is based on the NASA GSFC Software Engineering Lab's Process Improvement Paradigm that uses measurement as the basis for determining the effectiveness of our efforts to introduce improvements into V&V processes. In general, a process improvement model iterates over the following steps:

1. *Measure* the current process;
2. *Analyze* strengths and weaknesses;
3. *Improve* the process by developing and introducing new technologies to addresses weaknesses;
4. *Measure* the process to determine the effectiveness of the improvement;
5. Repeat steps 2, 3, 4.

Figure 1 depicts an overview of the V&V organization and research group in context of a development process. The next sections describe the aspects of measurement in the V&V process improvement model: cost effectiveness, trend analysis, and error detection.

3.1 Measuring Cost Effectiveness

What is the value of V&V to a project? If V&V finds errors early in a project's life-cycle, what are these worth in terms of cost avoidance to the project in the long-term? Several models of cost avoidance estimation have been proposed in the literature [12,13], but they are very general and many assume that errors are not caught by development until testing at the end of the development life-cycle. More sophisticated models exist, but they are specialized with respect to development and V&V processes.

We propose a framework that can be customized for specific projects to track the cost of fixing errors in each life-cycle phase. The framework is based on existing cost estimation models and provides an evolutionary approach to improving the accuracy of cost-savings estimates throughout the lifetime of a project. This assumes that the development process is cyclic because it affords opportunities for repeated phases on the same project. Fortunately, our experimental V&V projects have cyclic development processes that consist of multiple releases over an extended maintenance phase. It is anticipated that the projects will incur significant functional changes that must undergo cyclic development phases.

For example, if a number of major problems are uncovered during the first requirements analysis phase of V&V, the cost savings can be estimated based on existing models within a wide confidence range [1]. In the next iteration of the requirements phase, we can better estimate the cost savings based on knowledge of costs to fix errors in previous iterations of phases for that project. This allows for increased accuracy of estimates and confidence in V&V assessments.

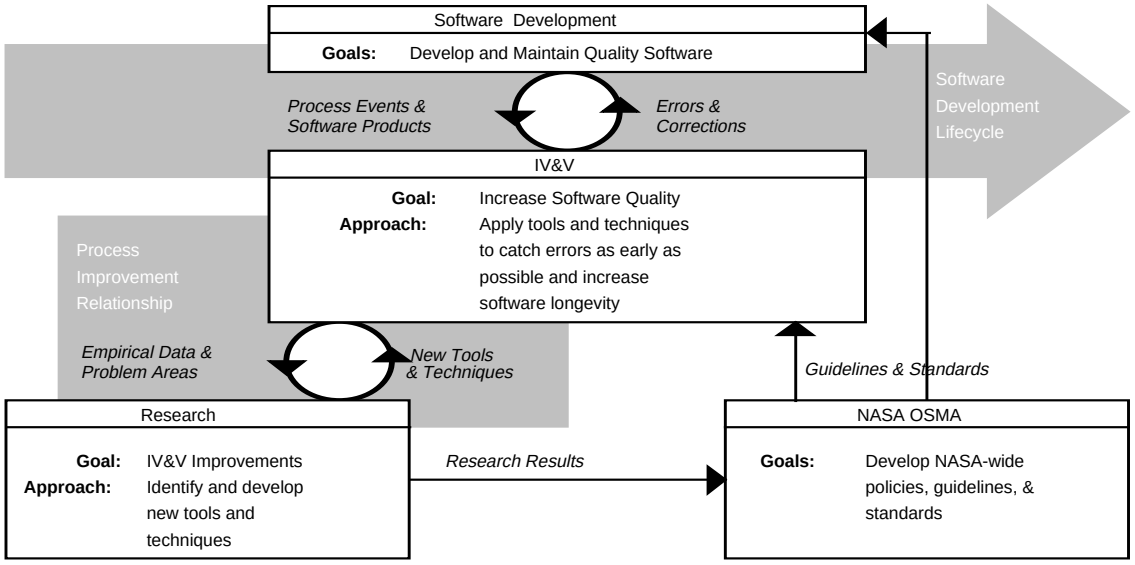

Figure 1: An overview of the V&V process improvement model

Part of our effort also involves factor analysis of V&V measurements to assess their impact on identifying potential problems. A V&V analysis may find problems, but these problems may be of high, moderate, or low impact. It is often difficult to assess the value of a technique at finding high-impact errors. More research is needed to identify effective techniques and incorporate them in V&V processes.

3.2 Trend Analysis

Can V&V help predict problems? The status of a project is more than the analysis of its parts. While the individual product errors may not be severe in a project, their cumulative effect can be serious. V&V efforts will yield analysis in the form of metrics on development processes and products. These metrics can be used by a V&V organization to predict trends that may result in schedule slippage, increased errors, costs, and other composite effects. It is necessary for a V&V organization to spot process problems early in the life-cycle and must have effective means to predict them. Our model relies on the cyclic phases of development to allow us to identify trends in software processes based on the analysis of correlation to find leading indicators in a project [14,15] that foreshadow potential problems. Once these indicators are identified and validated, they can also increase the accuracy of estimates and confidence in V&V assessments.

V&V has also been shown to have an influence on software reliability and maintenance. We are still modifying existing models to incorporate the ability to estimate the impact of V&V on reliability and maintainability. These qualities, however, are very difficult to quantify and only meaningful in the context of a project's goals. We are still exploring ways of quantifying such qualities in our model so that the full value of V&V on a project can be assessed.

As we identify improved V&V measurements and techniques, we will need to introducing new methods into the V&V life-cycle. Again, the cyclic nature of our associated projects allows for the incorporation of changes at strategic points in the process. Like the SEL model, our on-going measurements will allow us to assess the impact of such changes on the effectiveness of V&V.

3.3 Error Detection

How much and what types of V&V are required on a project? It is necessary to improve the ability of V&V to find problems in a software development project and focus analysis on the most critical aspects of development products and processes. Our framework will analyze the success and failure of existing V&V techniques to detect specific errors by auditing errors (i.e., V&V discrepancy reports) backward in the V&V process. Auditing these problems should help identify gaps in the V&V processes. For example, errors can be missed due to several problems in the V&V process including:

- *Omission*. The problem was caused by an error that could have been caught by the V&V process, but was overlooked due to the lack of V&V personnel expertise or the difficulty in applying the analysis;

- *Incompleteness*. The problem could have been avoided via existing techniques but the lack of information from the development process prevented its application;

- *Lack of Resources.* The problem could have been found but there was insufficient time or personnel needed to find it;

- *Lack of Capability*. The problem was caused by an error that could not have been caught by the V&V process because of the inadequacy of the methods and tools involved or the inherent complexity of the error.

This is not a complete list of reasons why errors are missed, but they are typical of the way in which errors can be classified in order to help improve detection of errors in earlier life-cycle phases. Analysis of classified V&V errors can lead to discovery of common types of errors that may suggest new methods, specifications, or processes.

4 Approach

The need to change V&V methods as part of an ongoing improvement program will impact the development process. For this and other reasons, much debate has surrounded the need for V&V. Some argue that it is more important to improve the quality of the development organization. It is beyond the scope of this paper to completely sort out the arguments, but we see the two views as compatible. A V&V should not simply assess the status of a development effort, but also provide feedback for improvement of the development process itself. In other words, V&V can act as a process improvement organization for development. The next sections describe our long-term strategy related to this view and our short-term tasks for achieving this goal.

4.1 Long-Term: Verifiable Development Techniques

Initially, we are focusing on the ability of the V&V process to find problems effectively and not on improving the capabilities of the software development process itself. However, because V&V and development are intimately related processes, we have developed a strategy for transferring improvements to development processes based on the need for improvements in V&V.

Our long-term strategy is to demonstrate that changes to development are needed in cases where V&V is unable to perform its task due to inappropriate or unavailable information from development. The goal of process improvement on a development organization is to enable it to produce high-quality software, on time, and within budget. This implies that the development effort is predictable and measurable. Ultimately, this will lead to development techniques that are highly amenable to V&V activities. We have labeled these *verifiable development techniques* (VDTs) to identify them as enabling effective V&V over

other approaches. A verifiable development technique is comprised of many different phases that are highly amenable to V&V. For example, the requirements for a safety-critical project might be expressed in specification language that is amenable to formal analysis. In a VDT, such analysis is not simply a spot check but coordinated with analyses performed in other phases.

4.2 Short-Term Tasks

Current research activities are focused on the short-term tasks to construct the V&V process improvement framework. The framework is needed to form the basis of any future improvements in the area of V&V. While it is true that V&V activities have been conducted on projects for many years, industry has yet to define and document V&V processes involved with any degree of consistency. Working with real projects using real project data gives our research effort the unique ability to define a baseline set of processes that can then be improved through use of a structured improvement process.

Many metrics, models, techniques and processes exist that can be incorporated into our framework. We must identify those that currently exist and attempt to formulate the characteristics of new approaches. Our short-term tasks related to our long-term vision include:

- *Metrics*. We have identified some metrics that are highly effective in predicting the potential occurrence of problems in software projects. We are paying particular attention to existing metric "success" stories and studies. In addition, we are examining the "Hawthorne effect" in software development that occurs when a V&V organization is employed. We are working with the NASA Langley SEES effort to establish V&V baselines and compare experimental results of employing V&V.

- *Processes*. We are examining existing development processes and determining how to map V&V processes to them. In addition, we are examining V&V as related to non-standard development processes, particularly in large-scale projects where requirements change dramatically during development.

- *Classification*. Because V&V cannot be applied uniformly across all phases and products due to resource limitations, we are seeking means to classify software products according to their impact on system failure. Such classification schemes will help tailor V&V processes to direct their attention to appropriate problems.

- *Testing*. This traditional role of V&V cannot be totally ignored, but we plan to move "testing" to earlier stages in the software development life-cycle. For instance, a "test" of the requirements specifications can be posed as a challenge to be disputed by some analysis on the project requirements. We are also exploring the possibility of evolving early tests into executable test suites.

Work in these areas will help establish the criteria for validating our framework employed on ongoing projects. They are needed to establish means of assessing the cost estimates and error detection methods at all phases of the development and V&V life-cycles.

4.3 Validation Through Application

The concept of "Strategic Alliances" formed between government, industry and academia plays a critical role in the process of validating research artifacts. The research strategy used at the IV&V Facility consists of working relationships between research and select projects and organizations. Potential prospects for collaboration are selected through initial discussions that focus on determining if there is some mutual interest to serve as a basis for the collaboration. The ability to gain access to an independent research organization that has the potential to improve processes and products without disrupting the

normal schedule of project activities is usually a very attractive incentive to induce project cooperation. It provides the project with research derived information and insight that would otherwise be absent. The only cost to the activity, in return, is to supply the research organization with "real" project data that is needed to corroborate their efforts.

Figure 1 also depicts the relationship described above. It describes the relationship between a developing agent, an IV&V agent, a research agent, and a governing body. However, the process could work just as well without an IV&V agent in which case research would interface directly with the developing organization. Both cases are in effect at the Facility and seem to offer equal benefit.

For each project, software quality is achieved through process improvement. First, one must define a starting point or baseline. If improvement is to be made we must know where we are at. This, in the case of the Facility is achieved by understanding the current practices of each of the selected projects or activities and using it as a baseline. Second, there must be a method by which to measure the improvements that are made. This can be accomplished using existing project metrics augmented by the introduction of any research specific metrics that may be needed. Third, an organization is needed whose focus is the introduction and measurement of new processes and products. This is the role played by the research organization, Fourth, there must be a governing body that is responsible not only to fund the improvement process, but to transform the results into usable products through establishment of policy, standards, and guidelines that in turn can be shared throughout the industry.

In this model, research plays a crucial role. A developing agent seldom has time allocated to explore potential improvement initiatives. Project cost and schedule matters are almost always take precedence over evolving technology. Access to a research organization whose charter is technology improvement allows advances to be made with a minimum amount of impact to the developing agent. Research in turn, benefits from the real-time validation it receives because results have been derived on real projects as opposed to projections based on theory and classroom trials.

4.4 A Case Study: EOSDIS

One example of this type of collaboration is our on-going work with the EOSDIS IV&V contractor to provide V&V process improvement on a long-term development project within NASA. The EOSDIS project is well-suited because it is still in its earliest development phases and open to collaboration. It is a large project with significant risks that can benefit from V&V because its development life-cycle is cyclic due to staged releases of program functionality and anticipated upgrades. We view this has a unique opportunity to introduce a process improvement model for V&V in order to ensure increasing confidence in the face of functional enhancement and a long-term maintenance phase.

It is still too early in the EOSDIS effort for substantive measurements, but initial audits of discrepancy reports generated by V&V suggest that a major obstacle is the lack of timely and appropriate products from the development organizations supplied to the V&V contractor. For example, project schedules were provided in Gantt chart form with little information about associated effort or context. Furthermore, the time allotted to V&V to analyze the schedule did not allow the application of cost and schedule estimation models. This limited the type and extent of V&V analysis on the development schedule.

The preliminary requirements analysis of the ECS portion of EOSDIS was completed at the end of October 1994. Currently, we are in the process of performing cost avoidance estimates on the preliminary requirements analysis and assessing the effectiveness of the analysis. The cost avoidance of errors found in this early phase will be estimated based on available models and later compared with actual performance. We will also produce confidence levels associated with these estimates.

There is also serious concern in the EOSDIS V&V effort over the fidelity project requirements and designs. While several errors were found in the requirements, it is questionable whether or not they are in

agreement with current design artifacts. The V&V contractor discovered this problem and the development contractor is currently fixing it before the start of the next V&V phase.

5 Summary

The NASA IV&V Facility was established in 1994 as part of a larger effort within NASA to focus attention on software issues. It currently houses efforts related to the Earth Observing System (EOS) and Space Station projects. It also houses a university research team committed to measurement-based research on actual V&V projects. This unique environment will create a testbed for new techniques in software product and process analysis.

Ultimately, we hope to improve the quality of computer software and the organizations that develop or help develop it. This paper does not seek to justify the use of V&V in projects but to (1) establish guidelines for determining its effectiveness and (2) improve its practice. By basing our work on a sound measurements program, we hope to frame V&V effectiveness within the context of its application. We hope that our process improvement model for V&V can benefit both V&V and development efforts.

Many barriers still remain to conducting research on software development and V&V efforts. First, many vendors are reluctant to provide measurements because it will expose them to criticism. Second, visibility into proprietary techniques and processes may harm their competitive advantage. Finally, measurements provided by the measured project will always tend to be skewed optimistically. We are trying to address these barriers through memorandums of understanding and other contractual mechanisms.

On large software efforts, several agencies of the US government, including NASA, have invested heavily in independent V&V as insurance against catastrophic errors. As development methods evolve, V&V processes must also improve. Since V&V is a complementary process, its improvement will drive improvements in development. We see the relationship as mutually beneficial in achieving high quality software.

References

[1] Lewis, R., Independent Verification and Validation: A Life Cycle Engineering Process for Quality Software, John Wiley & Sons, New York, 1992.

[2] The Cost-Effectiveness of Independent Software Verification and Validation, NASA Jet Propulsion Laboratory, 1985.

[3] Radatz, J., Analysis of IV&V Data, Final Technical Report, Rome Air Development Center, March 1981.

[4] Kosowski, E., Perspectives on Software Development and Verification - Boeing 757/767 AFDS, *Proceedings of the IEEE/AIAA 5th Digital Avionics Systems Conference*, IEEE, October 31 - November 3, 1983, pp. 6.5.1 - 6.5.4.

[5] Nicolai, R., Verification and Validation of IRAS On-Board Software, *Proceedings of the ESA/ESTEC Software Engineering Seminar*, ESA-SP-199, October 11-14, 1983, pp. 221-226.

[6] Daggett, P., M. Forshee, S. Forest, T. Fox-Daeke, G. Ingram, and D. Papa, *Handbook for Evaluation and Life-Cycle Planning for Software, Volume IV, Test and Independent Verification and Validation*, ESD-TR-84-171 (IV), 1983.

[7] Sapp, J. and C. Southworth, *Orlando I -- Final Report -- Panel B -- Independent Verification and Validation (IV&V)*, Joint Logistics Commander JPCG-CRM-CSM Conference, October 1983.

[8] McGarry, F., What have we learned in the last 6 years -- measuring software development technology, *Proceedings of the 7th Annual Software Engineering Workshop*, NASA/GSFC, December 1982.

[9] Brosius, D., *Software Validation Study*, SAMSO TR-73-99, 1973.

[10] National Research Council, An Asssessment of Space Shuttle Flight Software Development Processes, National Academy Press, Washington, D.C., 1993.

[11] NASA Office of Safety and Mission Assurance, *Proccedings of the Verification and Validation Workshop*, Morgantown, WV, December 1993.

[12] Boehm, B., *Software Engineering Economics*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.

[13] Wolverton, R., *Airborne Systems Software Acquisition Engineering Guidebook for Software Cost Analysis and Estimating*, ASD-TR-80-5025, Aeronautical Systems Division, 1980.

[14] Dyson, P., K. Dyson and J. McGhan, *Streamlined Integrated Software Metrics Approach (SISMA) Guidebook*, Software Productivity Solutions, Indiatlantic, FL, 1993.

[15] Callahan, J., T. Zhou and R. Woods, Software Risk Management Through Independent Verification and Validation, *Proceedings of the 4th International Conference on Software Quality*, American Society for Quality Control, Mclean, VA, October 305, 1994.